www.ijsrnsc.org

Available online at www.ijsrnsc.org

IJSRNSC

Volume-4, Issue-2, Apr- 2016
Research Paper

Int. J. Sc. Res. in Network Security and Communication
ISSN: 2321-3256

Key Management in Hierarchical Sensor Networks Using Improved Evolutionary Algorithm

G.R. Shahmohammadi[*1] and Kh.Mohammadi[2]

[1] *Department of information Technology, University of olum Entezami Amin, Iran*
[2]*Department of Computer System, Faculty of Engineering, University of Ashtian Islamic Azad, Iran*

Received: Mar/19/2016 Revised: Mar/28/2016 Accepted: Apr/17/2016 Published: Apr/30/2016

***Abstract*— Secure key exchange in wireless sensor networks is a topic always considered in terms of processing power and power consumption of sensor nodes in addition to security problems. Because of limited processing and energy in these nodes, energy consumption during this exchange process is one of the most important components examined in key exchange .In this paper, a hierarchical model of key distribution management is presented in which energy consumption is reduced in nodes, leading to increased network stability and longevity of nodes. In Wireless Sensor Network security field, each pre-distribution key approach demands appropriate management on keys, which is known as key management scheme. The main focus on the issue of key management is how to generate keys and control the power (energy) of nodes as well as lower the memory consumption.**

The proposed method is based on evolutionary algorithms. Firefly evolutionary algorithm has been used to find the public optimal solution as soon as possible, optimally solve the potential test functions as well as solving optimization problems in case of presence of noise in data. Thus, the first objective of this thesis is optimal production of safe keys in hierarchical sensor networks using a firefly algorithm. Evaluation criteria of the proposed algorithm include power (energy) control in nodes as well as lower memory consumption .The second objective of this study is to develop an improved firefly optimization algorithm in which improved synchronization rate and algorithm performance is sought via convergence of main and control parameters. Experimental results of the proposed method on different distributions of sensor nodes, header and sink prove the superiority of proposed method in comparison to the other key management schemes.

***Keywords*—Wireless sensor networks(WSN), Key management optimization algorithms, Evolutionary Algorithms, Firefly Algorithm.**

I. INTRODUCTION

A wireless sensor network consists of a large number of small sensor nodes without careful with limited processing capabilities and communications .The nodes away from the central node transfer their data through intermediate nodes and multi-hop protocols. In this case, the nodes may be both producers of information and data transmitters. The problem in such networks is that sensor nodes are normally fed by a battery and it is usually difficult or impractical to change or charge the battery due to high costs or use in inaccessible areas. Sources of energy consumption in sensor networks include sensing, data processing, Sleep and data transmission (including three modes of sending, receiving, and unemployment). Some of the major sources of energy loss in these networks can be categorized as follows: collision, crosstalk, control overhead packets, idle, non-optimum use of existing resources such as non-optimal routing to send and receive information and lofty transmit power in cases in which unnecessary special structure of these networks creates new problems for security in these networks. These problems can be considered a product of several factors, including wireless transmission environment, dynamic structure, absence of a fixed infrastructure, weaknesses related to networking nodes, large and dense networks, high risk of physical attacks and unknown network topology before developing methods for maintaining security in wireless sensor networks, which should at least provide for integrity authentication confidentiality, expandability and flexibility. Nowadays, key management is one of the methods to prevent attacks and ensure the security in these networks .Wireless sensor networks in recent years have been of interest for researchers although key management has its own specific problems and challenges [1].

In security terms of wireless sensor networks, each key pre-distribution method requires appropriate management on keys, which is known as key management .The majority of key management methods are mentioned with key pre-distribution methods with the same objective. In the field of key management, the principal focus is how to generate keys for two main reason [2]: power control of nodes and lower memory consumption.

Hierarchical sensor networks are a combination of sensor nodes, sink nodes and header nodes .These three types of

[*]Corresponding Author: G.R. Shahmohammadi
E-mail: shah_mohammadi@yahoo.co.uk

nodes in hierarchical sensor networks are different in terms of energy consumption, the amount of memory and computing capabilities. Hierarchical architecture significantly eliminates management problems on a network. For example, delegation of additional duties to the chairman of clusters significantly reduces network overhead because each of the tasks is done by the clusters' president individually [3].In addition, the hierarchical architecture is very suitable for data collection, which is done by the header .In a hierarchical architecture with designated security duties and routing for cluster's president, the number of operations to be implemented in the entire network for security and routing of the network is decreased and the head runs the security protocols and routing of each cluster according to their cluster.

In previous research studies, problems such as lack of consideration for energy consumption to increase network size, reduced security, low scalability, possibility of increasing the node and memory shortage to create more security have been addressed[2,4,5,6,7].In the proposed method, we have attempted to fix the problems and identified two objectives in this article: 1)optimal production of safe keys in the hierarchical sensor networks using firefly algorithm (FA), 2)creation of an improved FA in which the main parameters and controls have been synchronized to cure the convergence rate and the performance of our algorithm .Then, in the second section, related studies and in the third section, the firefly algorithm has been presented. The proposed method is described in Section IV. Finally, conclusions and future work are presented.

II. RELATED RESEARCH

Lawrence and Colleagues [2] offer a method to improve key pre-distribution considering average distance between nodes in Wireless Sensor Networks and showed that combining a main key pair in a randomized design reduces any possible average distance between nodes and increases network coverage.

Huang and colleagues [8] offered a key management method for heterogeneous Wireless Sensor Networks by all the nodes in the cluster package in which cluster heads produce key rings, sensor nodes and clusters together as a pair of keys. They use hash functions that reduce the memory for sensor nodes and also ensure the security key .Lai and Colleagues [9] provided a key pre-distribution scheme in which the main key is pre-distributed and stored in all pre-distributed sensor nodes. With this master key and transaction of a random number, each sensor node generates a key pair. This design is highly scalable and each node is limited in terms of memory .The problem with this method is that if the primary key is discovered by the influential factor, all the couple keys will be discovered. To improve this technique, Zhou and colleagues [10] presented a method in which the primary key disappears after a couple key is generated.

Wang [4] offered a scheme for key management based on genetic algorithm in hierarchical wireless sensor networks .The project is divided into three parts: well node, chairman node and sensor node. In the first phase, the sink nodes create key generation functions using genetic algorithm and then these functions are sent for well nodes and the sensor. Finally, the function of several keys is used to generate the keys again. Joint keys are combined together and generate a new key using the functions of key production so that the chairman and sensor nodes communicate securely. Although the key management has been cited in literature, the problem of related researches is that in [2], energy consumption in exchange for the increased network size has not calculated but we calculate energy consumption in the proposed method. In [8], the average number of reserve keys in the nodes is calculated for the number of sensor nodes and the number of stored keys is minimized in this layout, which reduces the memory space. The disadvantage of this method is that the amount of stored energy in exchange for the reduced number of keys has not been calculated. The problem in [11] is that there is a cluster head node sending a key generation function for the sensors, the sensors produce the same keys, and if a key is disclosed, all the keys are disclosed and network security comes down, and the entire network is compromised. The defect in [12] is that there is no security in this plan, and it will require an expensive intervention stable hardware with less elasticity and authenticity. The defect in [5] is that there is low scalability because the project base station needs to send a pair of keys to the sensor nodes. The major limitation in [13] is the integrity nodes that may be targeted nodes as well as possibility of an agreement to be reached by the enemy. The problem in [6] is that a sensor node is not allowed to add node to network because there is a node with no recent key pair, causing addition of sensor nodes to the network and lack of scalability because each sensor stores the keys as the total number of sensors stored in the network. In [7] the main problem is that security has increased and this increased security, greater memory consumption.

III. FIREFLY ALGORITHM

There are two important points in Firefly Algorithms: variation in light intensity and formulated glowing. The charm of a firefly glow is determined by the associated objective encrypted and indicated by β. The firefly glow is variable by changing the distance between I and j shown as r_{ij}. In addition, the light intensity decreases with distance from the light source and the light is absorbed by media and thus the shine changes with suction rate. The light intensity is shown as I (r). According to the inverse square law, it is shown as follows [14]:

1) I (r) $= \frac{I_s}{r^2}$

When the distance between the two objects is increased, light intensity is decreased between them. For a given environment with constant Light Absorption Coefficient of γ, the initial light intensity of I_0, and distance of r, the intensity of light is calculated by the following equation [14]:

2) $I=I_0 e^{-\gamma r^2}$

Sometimes we need a function uniformly reduced by a fixed rate. In this case [14]:

3) $I(r) = \frac{I_0}{1+\gamma r^2}$

Since the glow of Fireflies is proportional to the light intensity received by nearby fireflies and considering light intensity equation, the glow of each firefly is calculated by the following equation [14]:

4) $\beta(r) = \beta_0 e^{-\gamma r^2}$

Here, β_0 indicates the initial glow of firefly in [0, 1]range. If β_0=0, the search is random and there is no interactive relationship between the particles. In other words, each particle continues to search by itself. At the other extreme, if β_0=1, the search is done interactively (collectively) and locally. The r parameter is the distance between the two Fireflies [14]. The γ parameter is light absorption coefficient specifying the changing shine and its value varies by changing distance between the two fireflies. Gamma change can occur in two ways. In the first case when $\gamma \to 0$, the shine has the fixed value of $\beta = \beta_0$ and the light intensity will not increase as if there is no change or the shining is constant. A Brilliant worm can be seen anywhere in the domain. This causes a high rate of convergence in the algorithm [14]. In many cases, increased convergence results in placement of algorithm in an optimized position. In the second case, we have $\gamma \to \infty$. In this case, the glow from other worms is almost zero; for instance, when the fireflies randomly fly in a foggy atmosphere. None of the worms have vision, and each worm moves in a totally random way [14] like a completely random search method. The light absorption coefficient has a significant impact on the rate of convergence to improve its global optimization problems. Since $\frac{1}{1+r^2}$ is an exponential function calculation, the previous formula can be replaced with the following formula [14]:

5) $\beta(r) = \frac{\beta_0}{1+\gamma r^2}$

The distance between I and j worms at x_i and x_j points is a Euclidean distance as follows [14]:

6) $r_{ij} = \left\| X_i - X_j \right\| = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2}$

In the above formula, $x_{i,}$ (I, k) is the k_{th} dimension after the specific coordinates of x_i from the i_{th} firefly. The distance between I and j fireflies is calculated in the following formula [14]:

7) $r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$

The equation of motion of the particle I to the other particle (j) with more shine is calculated by equation (3-8) [14]:

8) $X_i = X_i + \beta_0 e^{-\gamma r_{ij}^2}(X_j - X_i) + \alpha(\text{rand} - \frac{1}{2}$

The second term from left is dependent upon shine while the third term produces a short random motion. Rand parameter generates a random number in [0, 1] range.α parameter is for a short random motion for fireflies which similar to ϒ has a significant impact on convergence rate to find a universal optimum [14]. FA algorithm is capable of simultaneously finding optimum global cure among several locals with an effective method. Another advantage of FA algorithm is that each worm can work almost independently and can thus be appropriate for parallel implementation. FA algorithms is better than bird procession (such as optimization of particle clusters) and Genetic algorithms since the worms are carefully assembled around the optimum particle and do not jump around [14].

IV. SUGGESTED METHOD FOR KEY MANAGEMENT IN HIERARCHICAL SENSOR NETWORKS

In this Section, a new approach is presented for key management in Sensor Networks hierarchy based on evolutionary algorithms. This algorithm is used for the following reasons, 1) finding the general optimal solution in the shortest time possible, 2) optimal solution of the possible test functions, 3) solving optimization problems in case of noise in the data [15], [16].Function generation keys designed by FA algorithm are used in the final step among the cluster nodes and sensor nodes under the constraint of energy consumption. As mentioned above, in hierarchical network, sensor nodes have lower energy compared to other nodes and have assumed the responsibility for monitoring the environment and collect information and send them for their header nodes. Header node shave more computing capabilities, more power and higher memory compared to sensor nodes. Header nodes will send data received from sensor nodes to sink node. It should be noted that access out of wireless sensor network is provided through sink node. The sink node establishes the key generation functions, which are used in later stages to generate the key. A key generation function is made up of a number of sections, and each section is a combination of an operand and an operator.

For example, compounds (codes) such as +3/4and+1 are examples of the sections of a key production function. Since these functions are produced at the sink node, if the well node selects all the three above-mentioned compounds, all the possible permutations of these three sectors makeup the key generation functions. After determination of the key generation functions by the sink node, m key generation functions are sent to the header nodes. It should be noted that the number of clusters used in the network has been set equal to m. Next, each header node generates a different key in order to communicate with each cluster member using the received key generation function .For example, a key generated using the above key generation is +1+3/4, which is equal to 1/75, and 1/75 is used as the link key between sensor node and cluster node .Different sets of codes generate different keys with unlike energy constraints. A suitable set of codes generates an even distribution of keys together with less energy consumption. In this paper, the standard entropy is used in order to assess key generation .The overall framework of the system proposed by this paper is divided into two parts: 1)search for the function keys (in the sink node); 2) key generation using key generation functions received from the previous section (in head node and sensor).In the first stage, a population of fireflies is generated, the number of fireflies is equal to q and the size per firefly is the sum total of operators multiplied by the number of operands. Each firefly offers a key generation function and the number of vintage keys for each function is equal to factorial of half size of firefly. After generation of an initial population of fireflies, the fireflies under FA algorithm are updated, and finally m key generation functions a reselected from among the best glow worm and are sent for cluster nodes .In the second part of the proposed system, the function keys received from the first part are used to generate keys in sensor and cluster nodes.

V. IMPROVED FIREFLY ALGORITHM

The Firefly has been improved in the following ways:

A) Synchronization of control parameters in firefly algorithm

As mentioned above, α and γ parameters have a constant value in firefly algorithm and have an important effect in finding the global optimal point in the standard firefly algorithm. These two parameters influenced the time complexity of firefly algorithm and the exact choice of their values affects the choice of global optimal and reduces the number of iterations. In early stages, the value of both α and Parameter's must be maximum because the search is randomly done at early stages due to lack of proper understanding of the issue. However, at the end generations, the value of both parameters should be minimized. The minimum value of γ parameters in the final steps will lead to algorithm integration toward global optimal. Minimum value for α in terminal generations causes improvement of synchronization indicators. If the value of α and γ is minimum and maximum in terminal generations, respectively, the optimal solution is jumped and increases the algorithm iterations. In order to synchronize γ and α parameters in each iteration, equations (9) and (10) are used, respectively:

9) $$\gamma(I) = \gamma_{max} - \frac{(\gamma max - \gamma min)}{NI} \times I$$

Each iteration to adapt γ, which uniformly reduces the value of γ at each iteration. In this equation, γ_{min} and γ_{max} parameters are minimum and maximum allowed values for γ.In this respect, NI variable specifiesthe total number of iterations and I the current iteration number [12-13].

10) $$\alpha(I) = \alpha_{max} + e^{\frac{Ln\left(\frac{\alpha min}{\alpha max}\right)}{NI} \times I}$$

Conformity of α parameter (10). In this equation, α_{min} and α_{max} are minimum and maximum values for α, respectively. Other parameters have been presented in the former equation. In firefly algorithm, each worm moves towards the brighter worms, which may cause the involvement of algorithm in the local optimal. In order to solve this problem and to escape from local optimum, the proposed algorithm has applied another improvement on the proposed algorithm of previous stage. In the proposed algorithm, the current global optimal position will be effective on the new position of firefly, and this overall improved situation will be updated in every generation. To apply this improvement, overall optimal position of firefly is substituted in to equation (11) as the motion equation of firefly until a new position of each firefly is achieved. Then, each worm is compared with global optimization as well as its neighbors and moves towards a brighter and more efficient neighbor. The following equation is used to add the position of the best firefly to motion equation of firefly:

11)
$$X_i = X_i + \beta_0 e^{-\gamma r^2_{i,gbest}}\left(X_{gbest} - X_i\right) + \alpha\left(rand - \frac{1}{2}\right)$$

In the above equation, $r^2_{i,gbest}$ shows the Euclidean distance between ith position of firefly and position of the best firefly (X_{gbest}).As it was stated, in the proposed method, the position of each worm is only compared with the position of the best firefly and therefore the comparison between each firefly with all other fireflies will be deleted.By eliminating the mentioned comparisons, computational complexity of the algorithm is greatly reduced. In addition, eliminating the mentioned comparisons reduces one of the repeat loops of firefly algorithm. The improved firefly algorithm is shown in Figure 1.

Require: HS parameter
1: Define the objective function f (xi)
2: Generate initial population of fireflies xi (I = 1, 2 ..., n);
3: Compute light intensity Ii for each xi using f (xi);
4: Define γ_{min} and γ_{max};

5: Define αmin and αmax;
6: while (I ≤ Max generation) do
7: Compute γ (I) using (5-1);
8: Compute α (I) using (6-1);
9: ForI= 1: n fireflies do
10: Compute distance between fireflies I and g_{best};
11: Move firefly I towards g_{best} firefly in d-dimension using (7-1);
12: Attractiveness varies with distance r via exponent[$-\gamma r$];
13: Evaluate new solutions and update light intensity;
14: end for
15: Rank the fireflies and find the current best;
16: end while
17: Post process results and visualization;
Figure 1.Improved Firefly algorithm

In the proposed system, string display is used for coding the key generation functions in fireflies**.** As mentioned, each firefly is considered a key production function including different combinations of operands and operators. Before firefly is produced and subjected to search process, a set of operands and operators are defined by sink node. In the proposed system, operands are positive integers and operators are a collection of operations such as addition, subtraction, multiplication, integer division and logical operations (like AND, OR). In the firefly, which is a key generation function, each function is made up of sections divided into operand and operator regions. In the following Figure, an example of a size 12 firefly is shown, which shows a key generation function with six operands and six operators.

+	7	-	1	*	9	/	3	AND	5	OR	2

Figure 2. An example of a firefly (key generation function) consisting of six operands and six operators

According to the above Figure, the key generation function includes six sections and the total number of keys equals 6! = 720**.** In the above example, 720 keys may include combinations that produce duplicate keys. Energy consumption is a measure with direct relationship with the number of keys generated in the cluster nodes. This article assumes that operations such as addition and subtraction consume one unit of energy, multiplication and division two units of energy and logical operations three units of energy. Priority of operators is such that the highest priority is related to multiplication and division operations and the lowest priority concerns logical operators but addition and subtraction operators have priority between multiplication and logical operators. As mentioned above, the cluster head and sensor nodes have limited energy consumption and since performing any operation is associated with energy consumption, key generation operations in the cluster head node should consume lower energy than energy consumption limit of cluster head node. The point that must be considered is that of the generation of simple keys spends less energy on cluster head nodes; however, the simple keys are easily detectable by hackers. Therefore, key generation functions must be designed in such a way that the keys are not easily discovered while consuming lower energy during key generation. It should be noted that increased energy consumption for key generation increases the key generation time. Therefore, if necessary, energy consumption can be calculated based on duration of key generation. Fitness function is used to determine the quality of the proposed solutions. In other words, the evaluation function will send feedback for improved firefly algorithm and represents the quality of the search for solution of choice. In this paper, entropy is a measure used and is known as Max Ent in papers [17-18].Suppose that Key is a discrete random variable of a possible system, the value of which is determined in key generation function. Moreover, if the key generation function includes m different sections, since the maximum number of keys for m sections equals m!, evaluation of m! Keys is impossible when the value of m is large. In order to solve this problem, n (n<<m) random keys are used to calculate the entropy measure. If a set of n keys are in the form of $\{key_1, key_2, key_3, \dots, key_n\}$ and the probability of each key is $\{p_1, p_2, p_3, \dots, p_n\}$, then:

12) $\sum_{i=1}^{n} p_i = 1, p_i \geq 0, i = 1,2,\dots,n$

Therefore, according to formulas (13), the entropy of each firefly (key generation function) is calculated as follows:

13) $$Entropy(firefly_i) = \sum_{i=1}^{n} p_i \log\left(\frac{1}{p_i}\right)$$

Based on the above criteria, increase in entropy will result in more uniform distribution of keys, which is a positive feature. So, increase in entropy will increase the quality of generated keys, and the firefly with maximum entropy is selected as the best key generation function. Therefore, the question of this paper is maximizing the value of evaluation function.

B) Key production stage

As previously stated, the proposed firefly algorithm is able to generate functions in the key manufacturing sector to produce keys consuming less energy, using lower memory to store sections of a function and finally lowering computational complexity to provide security. In this section, the key generation functions are used that were developed at the sink node by improved firefly algorithm. In this section, the cluster head nodes generate security keys using incoming functions and send them for their members. In the next step, the sensor node receives the key, encodes the sent packets using the received key and consigns it for its own cluster

head node. Accordingly, in a cluster, each sensor node has a key different from other sensor nodes, which unlike previous works (like [11]), increases the network security. In this paper, in order to prevent the penetration into sensor nodes, the very short time t is assigned for key generation in cluster head and sensor nodes. To determine the value of t, first the intrusion period to sensor nodes should be estimated and the t value should be assigned in a much lower value than the estimated one.

C) Laboratory results

The network to be tested is simulated in Matlab 2010 software. This network contains 51 nodes randomly distributed in a space of 600×500 square meters based on a normal distribution. Sensor nodes should be assigned to clusters, and clustering algorithms such as K-means should be used for this purpose. In this simulation, sensor nodes have been grouped to 10 clusters using K-Means. Among the entire nodes, one node is grouped as sink node, 10 nodes as head nodes and 40 nodes as sensors. It should be noted that the sink node has no limit for consumption of energy. The population of fireflies includes 100 fireflies, and the size of each firefly has been defined as the sum total of size of operators and operands. Given the existence of 6 operators and 6 operands in the suggested system, the dimensions of each firefly has been considered to be equal to 12. The set of operators defined within the system include the operations of addition, subtraction, multiplication, division as well as logical operators AND, OR and the operands are selected from the set of integers [0, 9].To evaluate the propose method, the results were compared to results of a number of similar methods used to manage and generate keys. In this comparison, the three proposed methods using evolutionary and meta-heuristic methods that are based on nature and have been used in wireless sensor networks have been compared and examined, including genetic algorithm (GA), Particle Swarm Optimization (PSO) and Firefly Algorithm (FA).GA is based on generation production features and includes a population of chromosomes, which is described in [19].

In PSO algorithm, optimization is based on the behavior of animals (category of birds and fish category) in nature. Setting of main parameters from improved Firefly algorithm, genetic algorithm, optimization of particles, FA and the space of problem have been presented in Tables 1 to 5. In Figure 3, the distribution of sensor nodes, amplifier nodes and well nodes have been displayed. In this Figure, sink node is indicated with black square, amplifier nodes with green stars and sensor nodes with blue stars. The red circles also indicated the confines of each cluster to the center of amplifier node.

Table 1. The values of the space parameters

Value	Parameter
51	Number of nodes
40	The number of sensor nodes
10	The number of amplifier nodes
1	The number of sink node
10	The number of clusters

Table 2. The parameter values of the proposed method

Value	Parameter
20	The size of the population
12	The length of each vector
0/01	The minimum amount of gamma
100	The maximum amount of gamma
0	The minimum amount of alpha
1	The maximum amount of alpha
100	The number of repetitions

Table 3. The values of the firefly algorithm parameters

Value	Parameter
20	The size of the population
10	The length of each vector
0/9	Gamma
0/4	Alpha
100	The number of repetitions

Table 4. The values of the genetic algorithm parameters

Value	Parameter
20	The size of the population
10	The length of each vector
0/8	Compound rate
0/1	Mutation rate
100	The number of repetitions

Table 5. The values of the optimization of particles parameters

Value	Parameter
20	The size of the population
10	The length of each vector
2	C1
2	C2
1	W
100	The number of repetitions

In Figure 4, the proposed method was compared with three other methods in terms of entropy standard. The horizontal axis of the graph shows total number of iterations and vertical axis represents the entropy standard entropy. As mentioned above, the purpose of the proposed system is to maximize the value of entropy. Increased value of entropy indicates that generation of keys has uniform distribution and thus different keys are generated with less redundancy and higher security .Based on this Figure, in all the methods, increased number of iterations and identifying the problem space increases the value of entropy. According to this graph, the proposed method has achieved maximum value of entropy in all iterations. After the proposed method, FA shows better results in the middle and final iterations relative to GA and PSO .It should be noted that according to Figure 4, FA has given the worst results in early iterations, but over time, a significant improvement in the results of this algorithm can be observed .The results of GA and PSO are similar in almost all the iterations .The reason for success of the proposed method is generation of keys with a uniform distribution in all iterations .In other words, synchronization of the firefly algorithm parameters causes the design of key generation functions in such a way to generate secure keys with minimum redundancy.

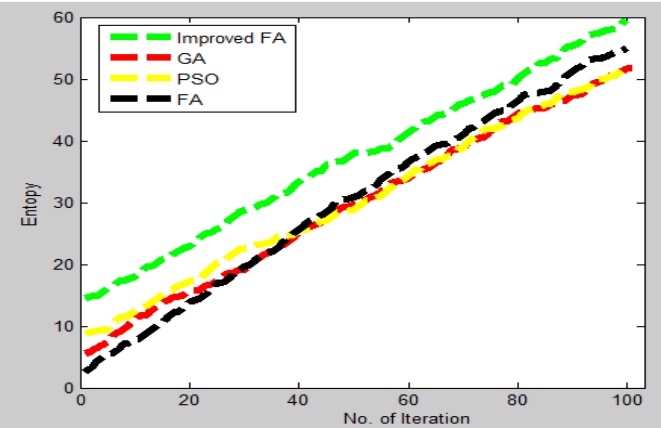

Figure 3. Distribution of sensor, amplifier and well nodes as cluster

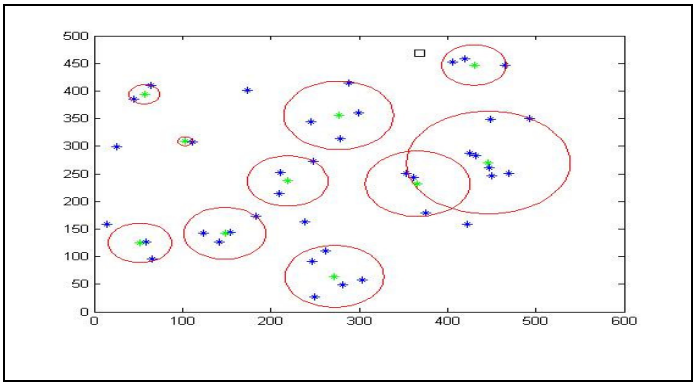

Figure 4. Comparison of the proposed method with other meta-heuristic and evolutionary methods regarding the entropy standard

In Figure 5, the proposed method was compared and evaluated with other methods in terms of energy consumption standard. As previously mentioned, the sensor nodes and headers are faced with restrictions on energy consumption. In other words, the sensor nodes and cluster head are faced with energy consumption issue in key generation process and the amount of energy consumed to generate the key must be lower than energy consumption limit of the nodes. The issue of energy consumption reduction at a particular node is realized when the key generation process is optimal. According to Figure 5, the proposed method shows minimum energy consumption to generate keys in the sensor nodes and cluster head compared with other methods. The reason for this is that the improved firefly algorithm directs the population to the best firefly in each iteration and optimally searches the problem space, designing key generation functions in a way that the lowest energy consumption will be achieved in cluster head nodes. The Particle Swarm Optimization algorithm is in the second rank. PSO can search the problem space and can offer optimum key generation functions. However, it may be stuck in a local optimum, which weakens the performances of this algorithm in optimum key generation and lower energy consumption. According to this Figure, the third rank of energy consumption is related to genetic algorithm, which does not show good results in comparison with the proposed method and PSO. Based on this chart, improved Firefly algorithm shows much better results compared to the conventional firefly algorithm. This reflects the impact of the improvements applied to the conventional firefly algorithm.

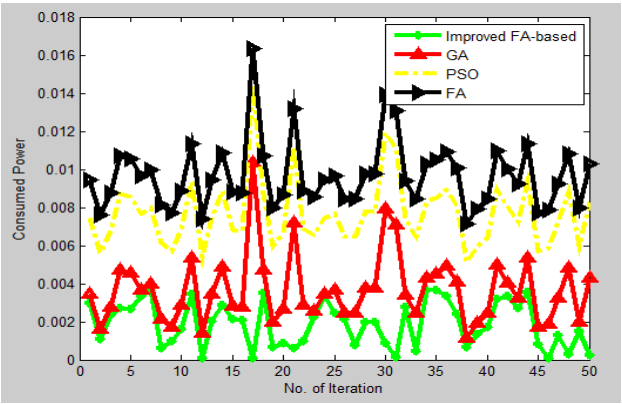

Figure 5. Comparison of the proposed method with other methods in energy consumption measure

The residual energy in the node is the opposite point of energy consumption. In key generation problem, the node consuming less energy for key generation will have higher residual energy to perform other operations, and thus many operations are performed in the node. In Figure 6, the residual energy in the cluster head node after key generation is the comparison benchmark between the proposed method and other meta-heuristic and evolutionary methods. Based on this Figure, the residual energy of the node would be

maximum when the proposed method is used to design the key generation functions. According to Figure 6, the particle optimization method has the second rank of residual energy in the node after the proposed method. In addition, genetic and firefly algorithms were ranked third and fourth in terms of residual energy.

In Figure 7, the proposed method has been evaluated in terms of key generation moment in the cluster and sensor nodes. This test has been used based on t time limit, which is defined at the cluster head nodes. The t time determines which cluster head node must be able to generate keys in this time limit. It should be noted that this time limit is much less than the capture time of sensor nodes by hackers. Thus, network security is increased by reducing the key generation time.

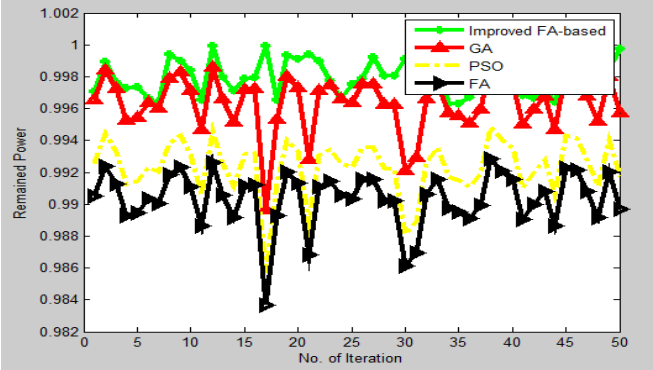

Figure 6. Comparison of proposed method with other methods for residual energy measure

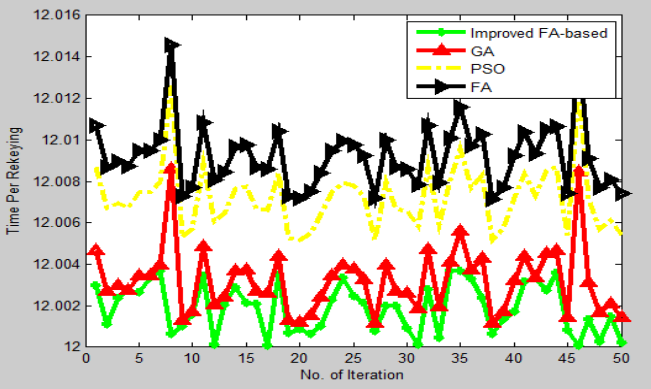

Figure 7. Comparison of the proposed method with other methods in terms of key generation time in the cluster head node

Furthermore, in Figures 8 and 9, the methods have been compared on the two criteria of the entropy and key generation time in the cluster head nodes when the number of cluster head nodes (the number of amplifier nodes) is variable.

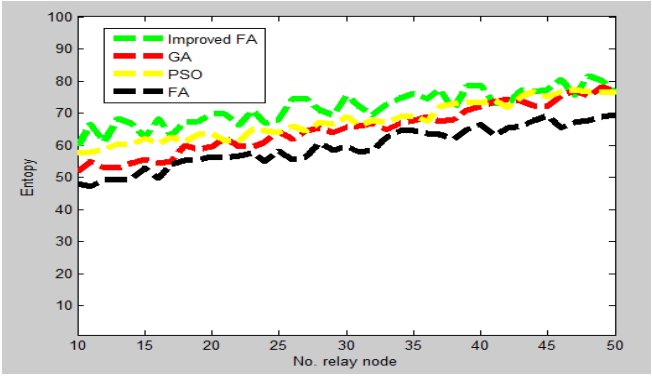

Figure 8. Comparison of the proposed method with other methods on the entropy criterion with variable number of amplifier nodes.

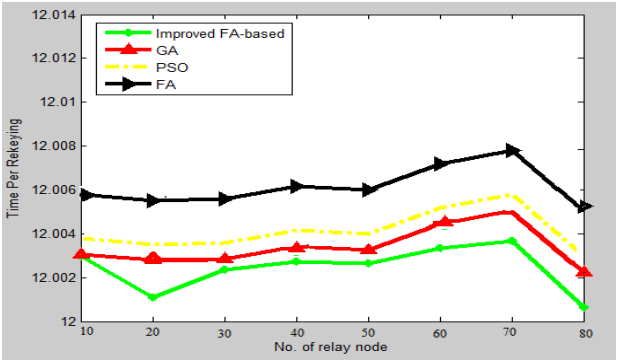

Figure 9. Comparison of the proposed method with other methods in key production time at the cluster head node with variable number of amplifier nodes

Based on the above experiments, the proposed method designed functions in the cluster head nodes such that minimum period of time was spent to generate keys. The reason for this is that the proposed method is capable of searching the problem space, generating functions that used a key production mechanism in which the cluster head node spent far less time in order to generate keys. In addition, the functions designed by the proposed method are simple and therefore less time is spent to generate keys using this method. It should be noted that the simplicity of functions is not indicative of insecurity of produced keys. In this experiment, optimization methods of PSO, FA and GA have been ranked second to fourth, respectively. Based on the results, improved firefly algorithm is the optimal method for key generation in hierarchy wireless sensor networks.

VI. CONCLUSIONS AND FUTURE WORKS

- In this article, the role and importance of sensor networks in energy consumption as one of the most important components evaluated in key exchange as well as problems in key management, including generation, storage, transfer and maintenance of key security have been dealt with. According to the existing

issues, including energy consumption, network stability and longevity of nodes, a new method of key management in hierarchical wireless sensor networks was proposed in this paper based on meta-heuristic algorithms. FA was the meta-heuristic algorithm used, which has been presented based on the behavior of fireflies in attracting each other in nature. The sensor network consisted of sink node, cluster head node and sensor node in which the well node has no restriction in d consumption of energy and memory compared to other nodes. The cluster head node has been used because of clustering the nodes in different groups and is considered as the bridge between the well node and sensor node. Cluster head nodes have the less constraints in the energy consumption and memory compared with sensor nodes. Sensor nodes have the highest limits on power consumption and memory, and are responsible for management and monitoring the network environment and data collection. Firefly algorithm in this article is improved in many ways and has been used to generate security keys in sensor nodes: Synchronization of control parameters of Firefly algorithm to improve the convergence rate

- Direction of population to the optimal route to find global optimal
- Reduced complexity of the algorithm by eliminating duplicate comparisons.

The proposed method is simulated in a wireless sensor network with 40 sensors nodes, 10 amplifier nodes and one sink node. It was evaluated in terms of entropy, energy consumption for key generation, residual energy after key generation and key generation time at the cluster head node. In comparison with PSO, FA and GA algorithm, it was shown that improved FA designs key generation functions in such a way that the lowest energy and time is consumed in the cluster head node to generate keys. The results of the proposed method in this article in comparison with previous results shows that our method has been able to significantly reduce energy consumption, cause uniform distribution of keys and reduce key generation time in cluster head nodes. In the continuance of this research, the following innovations can be applied on key management problem to improve consumption of energy and the time spent in hierarchically wireless sensor networks:

- Performing preprocessing on the initial population using evolutionary algorithms in order to improve the initial population.
- The evaluation of intrusion into the system when the key generation is performed in the sensor node.
- The routing method of encrypted packets in wireless sensor networks using the generated keys.
- Improving other evolutionary and meta-heuristic methods and their assessment in key management issue.

REFRENCES

[1] N. Ali, Sh. Javad, A. Fateme," Key exchange in wireless sensor networks for energy management approach", Tenth international conference Iranian security, pages 88-93, 2013.

[2] S. Lawrence, L. Qiaoliang, N. Mary and F. Bo, "Key Pre-Distribution And The Average Distance In Wireless Sensor Networks", Second International Conference On Computer And Network Technology(ICCNT), pp.212-216,China,2010.

[3] F. Etimad, V. C. Gungor, L. Nassef, N. Akkari, MG A. Maik, S. Almasri and I. F. Akyildiz. "A survey on wireless sensor networks for smart grid," Computer Communications, volume 71, pages22-33, 1november 2015.

[4] C.L. Wang, T.P. Hong, G. Hoing, W.H. Wang, "A GA-Based Key- Management Scheme in Hierarchical Wireless Sensor Networks", International Journal of Innovative Computing, Information and Control, Vol. 5, Number 12, pp. 4693-4702, December 2009.

[5] Perrig.A, Szewczyk. R, Wen.V, Culler.D, Tygar.J.D, "SPINS: Security Protocols for Sensor Networks", int,Journal of Wireless networks, volume 8, issue 5, pages 521-534, September 2002.

[6] Chan. H, Perrig. A, "Random key predistribution schemes for sensor networks". In: Proceedings of the 2003 IEEE symposium on security andprivacy, pages 197-213, May 2003.

[7] Du .W, Han .Y S, Chen .S, Varshney .P K, "A key management scheme for wireless sensor networks using deployment knowledge", Int, Conference of the IEEE Computer and Communications Societies, Volume 1, pages 586-597, April 2004.

[8] Y.J. Huang, I.E. Liao and H.W. Tang, "A forward authentication key management scheme for heterogeneous wireless sensor network", EURASIP Journal on Wireless Communications and Networking, Vol. 2011, No.6, January 2011.

[9] B. Walid, Y. Challal, A. Bouabdallah, and V.Tarokh, "A highly scalable key pre-distribution scheme for wireless sensor networks,"International Journal of Emerging Engineering Research and Technology, Volume 2, Issue 8, Pages 18-23, November 2014.

[10] A. Moshaddique, J. Liu, and K. Kwak. "Security and privacy issues in wireless sensor networks for healthcare applications." Journal of medical systems, Volume 36, Issue 1, pages 93-101, February 2012.

[11] C.-L. Wang, T.-P. Hong, G. Hoing, W.-H. Wang, "A GA-Based Key- Management Scheme in Hierarchical Wireless Sensor Networks", International Journal of Innovative Computing, Information and Control, Vol. 5, Number 12(A), pp. 4693-4702, December 2009.

[12] Lai. B, Kim. S, Verbauwhede. I, "Scalable session key construction protocol for wireless sensor networks", In: Proceedings of the IEEE workshop on Large Scale Real Time and Embedded Systems LARTES, pages 1-6, December 2002.

[13] Chan. H, Perrig. A, "PIKE: peer intermediaries for key establishment in sensor networks", in: Proceedings of the 24th annual joint conference of the IEEE computer and communications societies (INFOCOM'05), pages 524-535, March 2005.
[14] Nanda. S, Jagannath, and G. Panda, "A survey on nature inspired metaheuristic algorithms for partitional clustering", Swarm and Evolutionary Computation, Volume 16, pp. 1-18, June 2014.
[15] B. Kadri, D. Moussaoui, M. Feham and A. Mhammed an, "Efficient Key Management Scheme for Hierarchical Wireless Sensor Networks", Vol. 4, N0 6, pp. 155-161, June 2012.
[16] H. Xiaobing, M. Niedermeier & H. D. Meer, "Dynamic key management in wireless sensor networks: A survey", Journal of Network and Computer Applications, Vol.36, No.2, pp. 611-622, March 2013.
[17] M. Ito and M. Tanaka, "Localization of a Moving Sensor by Particle Filters", International Journal of Innovative Computing, Information and Control, Vol. 4, No. 1, pp. 165-173, January 2008.
[18] Ellis, R. Steven, "Entropy, large deviations, and statistical mechanics Entropy, large deviations, and statistical mechanics", Springer Science & Business Media, ISBN: ISBN: 978-1-46-13-8532-2, 2012.
[19] David. E. Goldberg, "Genetic Algorithm in Search, Optimization and Machine Learning", Kluwer Academic Publishers, Boston, MA, ISBN: ISBN: 0201157675, Pages 95-99, 1989.

Authors Profile

Gholamreza Shahmohammadi received his Ph.D. degree from Tarbiat Modares University (TMU, Tehran, Iran) in 2009 and his M.Sc. degree in Computer Engineering from TMU in 2001. Since 2010, He has been Assistant Professor at the Department of Information Technology, Olum Entezami Amin University (Tehran, Iran). Amin University (Tehran, Iran). His main research interests are Software Engineering, Software Architecture, Software Metrics, Software Cost Estimation and Software Security.

Khatereh Mohammadi is Computer engineering master student Department of Computer Engineering Islamic Azad University of Ashtian, Iran. Her main rese arches interests are Sensor Network and Network Security.

